SI Appendix for: Determining protein structures by combining semi-reliable data with atomistic physical models by Bayesian inference

Justin L. MacCallum, Alberto Perez, Ken A. Dill

May 4, 2015

Contents

Table S1: Backbone RMSD (Å) and cluster population for the three most populous clusters for each system.

	Cluster 1		Cluster 2		Cluster 3	
Target	RMSD	Pop	RMSD	Pop	RMSD	Pop
Ubiquitin	1.0	0.90	3.6	0.05	1.4	0.02
Protein G	2.8	0.42	0.9	0.13	2.5	0.12
Crystallin	3.5	0.28	3.5	0.16	2.8	0.14
Lysozyme	3.6	0.13	5.1	0.10	4.0	0.09
Thioredoxin	2.6	0.42	3.0	0.32	2.8	0.15
Ras	3.0	0.76	3.3	0.12	3.5	0.09
CheY	4.3	0.37	3.2	0.33	2.8	0.14
Calponin	4.9	0.76	7.1	0.12	5.5	0.06

1 Supplemental Results

1.1 Cluster populations

Table S1 shows the RMSD and population for the top three clusters for each system.

1.2 Huber et al Ensemble

Figure S1 shows the 20 lowest energy (of 200 total structures of ubiquitin produced by CYANA based on ILV-methyl labeled NMR data and TALOS predicted backbone torsion angles. CYANA produces a broad ensemble, where even most of the 20 lowest-energy structures are far from native.

1.3 Funneled energy landscape

Figure S2 shows that MELD produces a highly funneled energy landscape. Replicas at the top of the ladder explore a broad basin of non-native structures. Replicas at the bottom sample mostly near native structures (see inset).

1.4 MELD energy versus RMSD

Figure S3 shows that at the lowest temperature, the MELD energy does not strongly correlate with the RMSD. There are many alternative structures that are in good agreement with the experimental data.

1.5 Results from EvFold

Table S2 shows that MELD produces superior structures to the CNS-based EvFold pipeline using the same predicted contacts.

Figure S4 shows the energy of the native structure as a function of the active fraction and the distance cutoff. For all systems, the parameters used in this

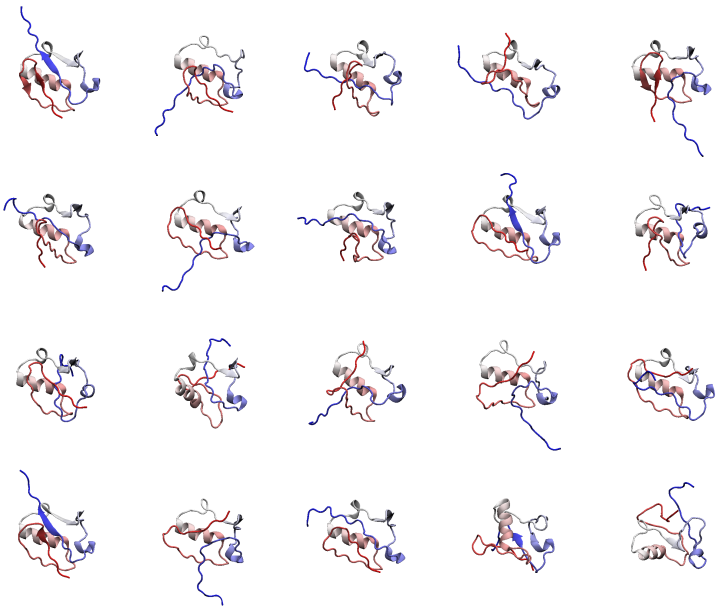

Figure S1: Lowest energy (20 of 200) structures produced with CYANA[1] coupled with predicted backbone torsion angles from Talos+[2] and short-range proton-proton distance information from ILV-methyl-labeled samples[3]. On average, the structures are not near-native, have poor secondary structures, and limited hydrogen bonding.

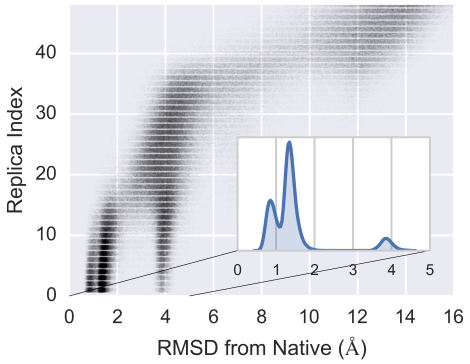

Figure S2: Conformational landscape in MELD simulations of ubiquitin using solid-state NMR data, showing a clearly funneled energy landscape. The inset shows the RMSD distribution at the lowest replica, demonstrating that most structures are below 1.5Å RMSD.

Table S2: Backbone RMSD (Å) between native and models produced by MELD or EvFold.

	MELD		EvFold	
Target	Best	Best Clust	Best	Lowest E
Thioredoxin	1.5	2.6	3.7	4.7
Ras	2.5	3.0	3.4	3.6
CheY	1.8	3.2	4.0	5.4
Calponin	4.3	4.9	5.2	11.4

study lie somewhat outside of the zero-energy region. However, all but calponin are close. Calponin would require a large reduction in active fraction or a large increase in distance cutoff to reach the zero-energy region.

2 Details of MELD

MELD generates a minimum free energy ensemble, subject to sparse, ambiguous, and probabalistic restraints. The MELD energy function can be explained in therms of three concepts: restraints, groups, and collections.

2.1 Restraints

MELD has a variety of different restraint forces, including flat-bottom harmonic restraints on distances or torsion angles, spline-based restraints on distances, and bicubic spline-based restraints on pairs of torsion angles. The specific functional forms are given in Section 4. Importantly, restraints in MELD are always

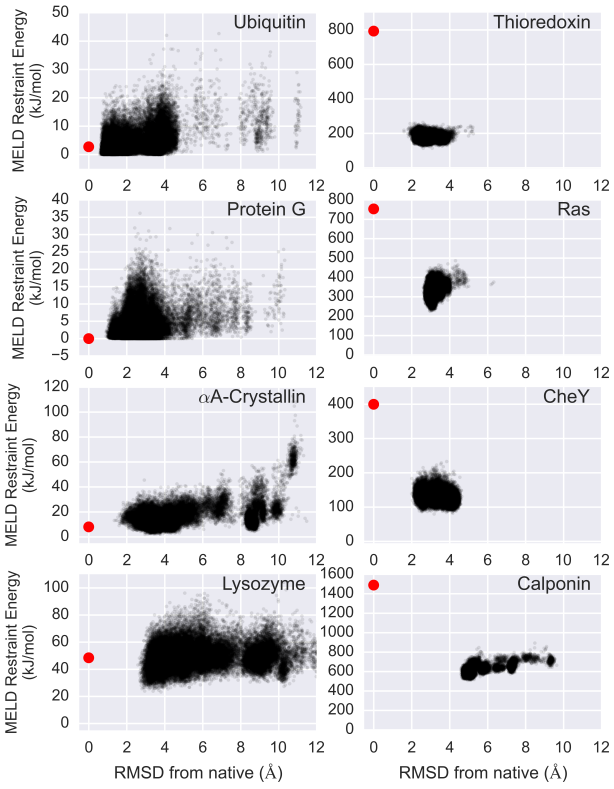

Figure S3: Each panel shows the MELD energy versus RMSD. In most cases, there is little correlation between the MELD energy of the sampled structures and the RMSD to native, which indicates that there are many possible conformations that are in good agreement with the experimental data. The red circle shows the native structure, which is discussed in the "enforcing incorrect restraints reduces accuracy" section of the main text.

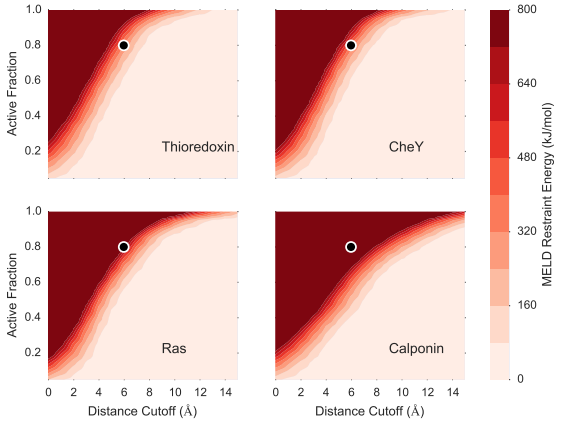

Figure S4: MELD energy of the native structure as a function of the active fraction of predicted contacts and the distance cutoff used to define a contact. The black dot indicates the parameters used in this study.

defined to be non-negative, so $E_{\text{meld}} \geq 0$.

2.2 Groups

Groups aggregate multiple restraints in collective, multi-body restraints such that only a specified fraction of restraints are activated. Each restraint must belong to exactly one group. The energy of group i is:

$$E_i^{\text{grp}} = \sum_{j=1}^{n_{\text{active,i}}^{\text{grp}}} E_{i,j}^{\text{rest}}, \tag{1}$$

where the component restraints are sorted by energy:

$$E_{i,1}^{\text{rest}} \leq E_{i,2}^{\text{rest}} \leq \cdots \leq E_{i,N_i}^{\text{rest}}. \tag{2}$$

The $\{E_{i,j}^{\text{rest}}\}$ are the energies of the component restraints that makeup group i and $n_{\text{active},i}^{\text{grp}}$ is a user-specified parameter that controls how many restraints are active from group i.

2.3 Collections

Collections are analogous to groups and provide a second level of sorting and activation to handle ambiguity and uncertainty. Groups combine restraints into collective terms, while collections combine groups into collective terms. Each restraint must belong to exactly one group, whereas each group must belong to

exactly one collection. The energy for collection i is:

$$E_i^{\text{coll}} = \sum_{j=1}^{n_{\text{active,i}}^{\text{coll}}} E_{i,j}^{\text{grp}}, \tag{3}$$

where the component groups are sorted by energy:

$$E_{i,1}^{\text{grp}} \leq E_{i,2}^{\text{grp}} \leq \cdots \leq E_{i,N_i}^{\text{grp}}. \tag{4}$$

The $\{E_{i,j}^{\text{grp}}\}$ are the energies of the component groups that makeup collection i and $n_{\text{active},i}^{\text{coll}}$ is a user-specified parameter that controls how many groups are active from collection i.

The total MELD energy in the simulation is:

$$E_{\text{meld}} = \sum_{i=1}^{N_{\text{coll}}} E_i^{\text{coll}}. \tag{5}$$

2.4 Guarantees of MELD

Consider two sets of conformations: the native conformations ($\mathbf{x} \in N$) and some other arbitrary set of conformations ($\mathbf{x} \in X$). If we examine the relative populations of these two states, both with and without MELD restraints, we have:

$$\begin{aligned} R &= \frac{p_{\text{meld}}^N}{p_{\text{meld}}^X} \frac{p_{\text{amber}}^X}{p_{\text{amber}}^N} \\ &= \frac{\int\limits_{\mathbf{x}\in N} e^{-\beta[E_{\text{amber}}(\mathbf{x})+E_{\text{meld}}(\mathbf{x})]} d\mathbf{x}}{\int\limits_{\mathbf{x}\in X} e^{-\beta[E_{\text{amber}}(\mathbf{x})+E_{\text{meld}}(\mathbf{x})]} d\mathbf{x}} \frac{\int\limits_{\mathbf{x}\in X} e^{-\beta[E_{\text{amber}}(\mathbf{x})]} d\mathbf{x}}{\int\limits_{\mathbf{x}\in N} e^{-\beta[E_{\text{amber}}(\mathbf{x})]} d\mathbf{x}}. \end{aligned} \tag{6}$$

If the native basin is compatible with the data, then by definition $E_{\text{meld}} = 0$ for all $\mathbf{x} \in N$, and Eq. 6 simplifies to:

$$R = \frac{\int\limits_{\mathbf{x}\in X} e^{-\beta[E_{\text{amber}}(\mathbf{x})]} d\mathbf{x}}{\int\limits_{\mathbf{x}\in X} e^{-\beta[E_{\text{amber}}(\mathbf{x})+E_{\text{meld}}(\mathbf{x})]} d\mathbf{x}}. \tag{7}$$

By construction $E_{\text{meld}} \geq 0$, which implies $R \geq 1$.

This result provides a strong guarantee. If E_{meld} is constructed appropriately—that is, the restraints are combined into group and collections with $n_{\text{active},i}$'s set appropriately—then E_{meld} will be zero for the native basin and we are assured that the population will go up compared to with the force field alone. This result also provides a strong constraint. If $E_{meld} > 0$ for the native basin, then we have no guaranties about populations and we are less likely to produce the correct structures.

Ensuring that $E_{meld} = 0$ for the native basin generally comes down to setting the $n_{\text{active},i}$ appropriately for each group or collection. The value of $n_{\text{active},i}$ for a group determines how many of the component restraints MELD must "believe". Ideally, the value of $n_{\text{active},i}$ will correspond exactly to the number of correct restraints. If $n_{\text{active},i}$ is set too high, the system will be forced to believe data that is wrong. If $n_{\text{active},i}$ is set too low, then we are ignoring data that could be providing valuable information. In practice, we set $n_{\text{active},i}$ based on past experience with that particular source of data.

2.5 Replica Exchange

The introduction of MELD restraints creates large barriers in the potential energy (Figure 1 in main text). Simulations at ambient temperature would quickly become trapped in a single basin, producing non-ergodic sampling. To overcome these barriers, we use Hamiltonian replica exchange (H-REMD)[4] to sample conformations.

MELD uses a 1-dimensional Hamiltonian exchange "ladder", where we vary both the temperature and the force constants of the MELD restraints. We define a parameter, $\alpha \in [0, 1]$, which varies along our ladder. The lowest replica always has $\alpha = 0$ and the highest $\alpha = 1$. Initially, the other replicas are spaced linearly. Each parameter that can vary (e.g. temperature or force constant) is expressed as a function of α (e.g., $T(\alpha)$, $k(\alpha)$). The value of α at each replica thereby determines the value of the parameters. During simulation, an iterative procedure occasionally adjusts the values of α at intermediate replicas to obtain equal acceptance rates between all pairs of adjacent replicas.

We typically divide the interval $[0, 1]$ in half. The temperature varies geometrically between 300K and 450K over $\alpha \in [0, 0.5]$, while the weight of the MELD restraints varies from 1.0 to 0.0 over $\alpha \in [0.5, 1.0]$. We find that this scheme provides good conformational sampling. Only the structures that are compatible with both the MELD restraints and the underlying force field are able to reach the lowest replica, which is clustered to determine the lowest-free-energy conformational basins[5, 6].

Overall, this approach is similar to simulated annealing strategies used in NMR structure determination [7]. However, unlike simulated annealing H-REMD produces Boltzmann distributions, where the population of a set of structures is related to its free energy. This allows MELD to select structures based on *free energy* rather than *energy*.

3 Simulation parameters

All molecular dynamics simulations were performed using a version of the OpenMM [8] simulation package modified to include the MELD forces. All simulations were performed using a modified version of the forthcoming Amber ff14sb force field (Carlos Simmerling, personal communication). The force field was modified by adding CMAP like corrections[9] in order to correct the balance between the helical and extended regions in implicit solvent. We use the GB model of

Onufriev, Bashford, and Case [10] to represent the solvent implicitly. Langevin dynamics simulations were carried out with at 2 fs time step and replica exchanges were typically attempted every 10 ps.

4 Functional forms of MELD restraints

MELD currently supports four different types of restraints. Extension to add new types of restraints is straightforward.

4.1 Harmonic distance restraints

The distance, r_{ij}, between two atoms i and j, can be restrained using flat-bottom harmonic restraints of the form:

$$E(r_{ij}) = \begin{cases} \frac{1}{2}k(r_1 - r_2)(2r_{ij} - r_1 - r_2) & \text{if } r_{ij} < r_1 \\ \frac{1}{2}k(r_{ij} - r_2)^2 & \text{if } r_1 \leq r_{ij} < r_2 \\ 0 & \text{if } r_2 \leq r_{ij} < r_3 \\ \frac{1}{2}k(r_{ij} - r_3)^2 & \text{if } r_3 \leq r_{ij} < r_4 \\ \frac{1}{2}k(r_4 - r_3)(2r_{ij} - r_4 - r_3) & \text{if } r_4 \leq r_{ij}, \end{cases} \tag{8}$$

where r_1–r_4 are distance cutoffs delineating the linear, quadratic, and flat regions of the potential, and k is the force constant.

4.2 Spline-based distance restraints

The distance, r_{ij}, between two atoms i and j can be restrained using a cubic-spline-based potential. The domain of $E(r)$ is divided into a series of piecewise cubic regions. First, we find the region, n, for the current value of r_{ij}. Next, we compute a parameter $t \in [0, 1]$:

$$t = \frac{r_{ij} - r_{\text{max},n}}{r_{\text{max},n} - r_{\text{min},n}}. \tag{9}$$

The energy is defined as:

$$E(t) = a_{0,n} + a_{1,n}t + a_{2,n}t^2 + a_{3,n}t^3, \tag{10}$$

where the $a_{k,n}$ values are parameters for region n, which are required to give C^2 continuity at the boundary between regions.

4.3 Harmonic torsion restraints

The torsion angle, ϕ, between four atoms (i, j, k, l) can be restrained using flat-bottomed harmonic restraints.

$$E(phi) = \begin{cases} \frac{1}{2}k(\phi' + \Delta\phi)^2 & \text{if } \phi' < -\Delta\phi \\ \frac{1}{2}k(\phi' - \Delta\phi)^2 & \text{if } \phi' > \Delta\phi \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where

$$\phi' = (\phi - \phi_0) \rightarrow (-180, +180] \tag{12}$$

is the difference between the current and desired angles modulo 180 degrees, ϕ_0 is the desired angle, and $2\Delta\phi$ is the width of the zero energy region around ϕ_0.

4.4 Bicubic-spline-based torsion pair restraints

A pair of torsion angles, ϕ and ψ, can be restrained using a bicubic-spline-based potential similar to the CMAP potential[9]. This potential is typically placed on the backbone ϕ/ψ angles for an amino acid. First, ϕ and ψ are converted into grid indices i and j. Then, the energy is calculated as:

$$E(\phi, \psi) = \sum_{i=1}^{4} \sum_{j=1}^{4} C_{ij} \left(\frac{\phi - \phi_{min,i}}{\Delta\phi} \right)^{i-1} \left(\frac{\psi - \psi_{min,j}}{\Delta\psi} \right)^{j-1}, \tag{13}$$

where C_{ij} are the spline coefficients, $\Delta\phi$ and $\Delta\psi$ are the widths of the grid cells, and $\phi_{min,i}$ and $\psi_{min,j}$ are the edges of grid cell i, j.

5 Data used in calculations

Figures S5–S12 summarize the data used in each case study. The upper portion of each panel shows the experimental structure and any distance restraints used in the calculation. The lower panel shows the predicted (upper band) and actual (lower band) secondary structures, along with the distance restraints. All possible Cα pairs with $|i - j| > 10$ and $||r_i - r_j|| < 10$ Å in the native structure are shown in grey. Correct distance restraints are shown in blue; incorrect distance restraints are shown in red. For the two EPR datasets (αA-Crystallin and T4 Lysozyme) restraints are colored depending on the measured distance; short distances (< 12 Å) are blue, long distances (> 12 Å) are cyan.

5.1 Treatment of secondary structure predictions

All secondary structure restraints in this work were based on predictions from PSIPRED[11]. We break the protein into $(N_{\text{res}} - 4)$ overlapping 5-residue fragments. If 4/5 or 5/5 of the residues in a fragment are predicted in state `H` or `E`, then we apply secondary structure restraints as described below. For all other fragments, we apply no restraints.

If the fragment is predicted to be likely helical or likely extended, we apply the following restraints (with local residue numbering from 1–5): backbone ϕ angles of residues 2–5, backbone ψ angles of residues 1–4, and distance restraints between the Cα atoms of residues (1,4), (2,5), and (1,5). The parameters for all restraints are given below. These 11 restraints (3 distances and 8 torsions) are combined into a single group with all 11 restraints active, so that they behave as a single "secondary structure fragment" restraint. All of the secondary structure restraints are combined into a collection with an active fraction of 0.75, which allows some 5-mers to differ from their predicted secondary structures.

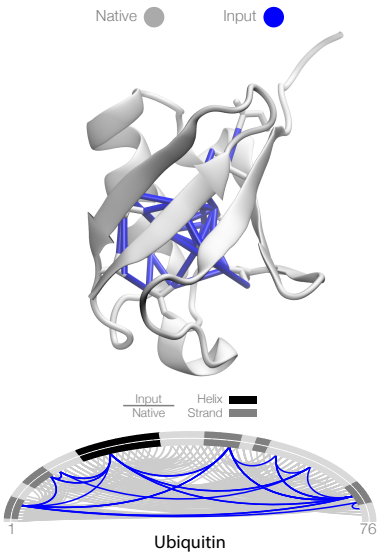

Figure S5: Data used in Ubiquitin calculation.

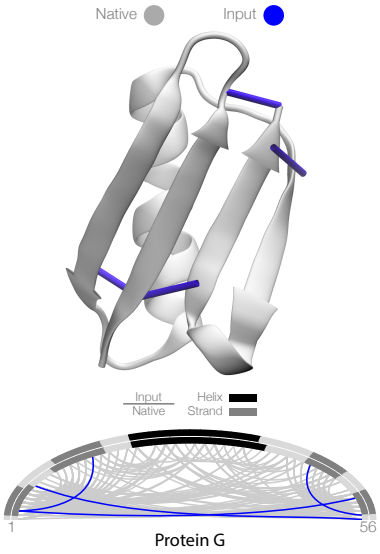

Figure S6: Data used in Protein G calculation.

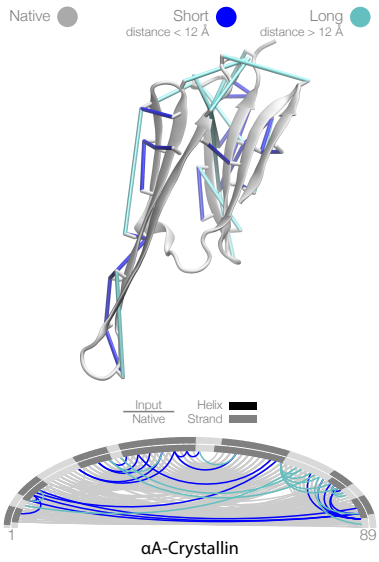

Figure S7: Data used in αA-Crystallin calculation.

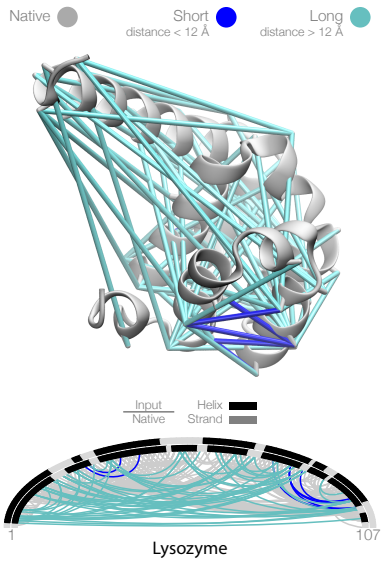

Figure S8: Data used in T4 Lysozyme calculation.

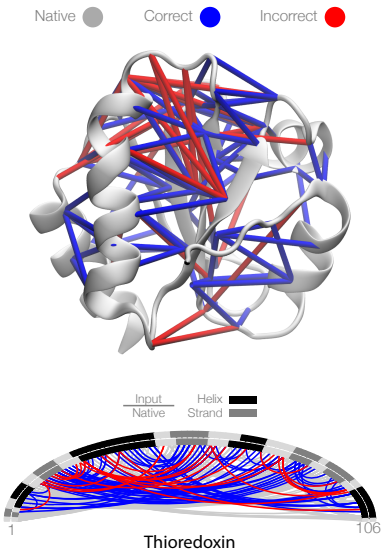

Figure S9: Data used in Thioredoxin calculation

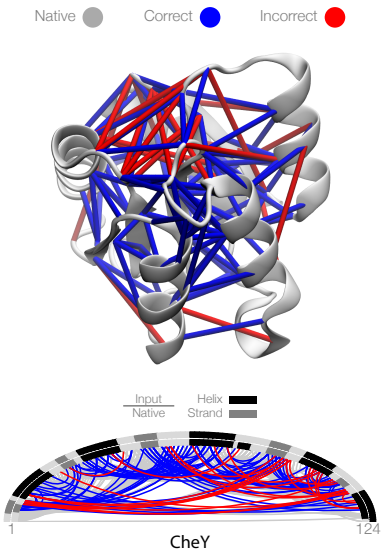

Figure S10: Data used in CheY calculation.

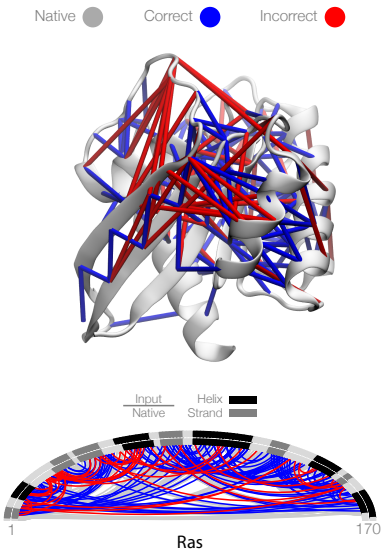

Figure S11: Data used in Ras calculation.

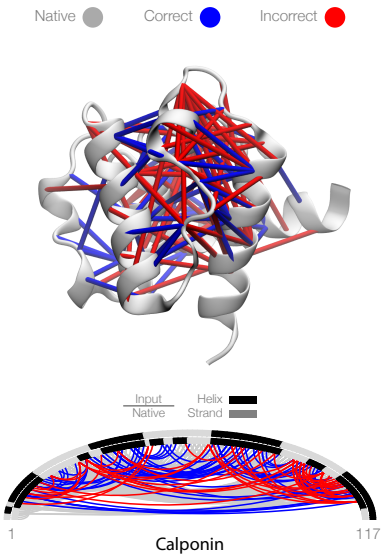

Figure S12: Data used in Calponin calculation.

The parameters for predicted helices are:

- ϕ between -62.5 ± 17.5
- ψ between -42.5 ± 17.5
- torsion force constant of 2.5 kJ mol^{-1} $(10°)^{-2}$
- 1–4 distance between 4.85 Å and 5.61 Å
- 2–5 distance between 4.85 Å and 5.61 Å
- 1–5 distance between 5.81 Å and 6.84 Å
- distance force constant of 2500 kJ mol^{-1} nm^{-2}.

The parameters for predicted strands are:

- ϕ within 117.5 ± 27.5
- ψ within 145 ± 25
- torsion force constant of 2.5 kJ mol^{-1} $(10°)^{-2}$
- 1–4 distance between 7.85 Å and 10.63 Å
- 2–5 distance between 7.85 Å and 10.63 Å
- 1–5 distance between 10.86 Å and 13.94 Å
- distance force constant of 2500 kJ mol^{-1} nm^{-2}.

5.2 Treatment of residue–residue information

We divide the treatment of residue–residue information into three categories: sparse, ambiguous, and probabilistic.

Sparse: this information informs about short residue–residue distances. This information is sparse, but otherwise unambiguous and reliable. The information is turned into flat-bottomed harmonic distance restraints and combined into a single group with all restraints active.

For the Huber solid-state NMR dataset, the restraints were between the terminal methyl groups of Ile, Leu, and Val residues. The stereospecific proton–proton interactions were mapped onto interactions between the corresponding carbon atoms. The energy was zero at separations up to 7Å. The energy increases quadratically from 7 to 8Å and then linearly beyond 8Å, with a force constant of 1000 kJ mol^{-1} nm^{-2}.

For Protein G, restraints were added between the Cα atoms of author selected residues: (3, 51), (9, 56), (3, 18), (42, 55). The energy was zero at separations up to 6Å, increasing quadratically from 6 to 7Å, and then linearly beyond 7Å, with a force constant of 1000 kJ mol^{-1} nm^{-2}.

All restraints were scaled from full-strength to zero over the interval $\alpha \in [0.5, 1]$.

Ambiguous: this information comes from site-directed spin label EPR experiments. It is sparse and measures a wide variety of distances with large uncertainties. We use the motion-on-a-cone knowledge-based energy model of Hirst *et al* [12] to turn each measured distance into distance restraint that reflects the uncertainty of mapping the experimental data into a geometric feature of the protein. We use either Equations 14 and 15 or a knowledge-based potential. The restraints are combined into a single group with all restraints active.

When using Equations 14 and 15, the energy was zero within d_{lower} and d_{upper}, increasing quadratically 1Å outside of these bounds, and linearly thereafter, with a force constant of 250 kJ mol^{-1} nm^{-2}.

For αA-Crystallin, we used the strategy from Ref. [13], where we set the allowed upper and lower bounds between Cβ atoms to:

$$d_{\text{lower}} = d_{\text{measured}} - \sigma_{\text{measured}} - 12.5\text{Å} \tag{14}$$

and

$$d_{\text{upper}} = d_{\text{measured}} + \sigma_{\text{measured}} + 2.5\text{Å}, \tag{15}$$

where d_{measured} is the measured probe–probe distance from Ref. [13], σ_{measured} is the measurement uncertainty, and 12.5Å and 2.5Å are constants based on a motion-on-a-cone model [13] of the spin-label probes.

For T4-Lysozyme, we followed the approach of Ref [12] and used a knowledge-based potential (also based on the motion-on-a-cone model) to represent the uncertainty in the Cβ–Cβ distance due to the flexible probe (Figure 2E of Ref [12]), which we implemented using cubic splines. The measured distance restraints are from Refs. [13] and [14].

Calculations using the knowledge-based approach were conducted as follows. The knowledge-based potential was digitized from Figure 2D of Ref. [12]. The experimental data was modeled as a Gaussian, with a mean and variance determined from the experimental data. This Gaussian was convolved with the digitized knowledge-based potential on a grid from 0 to 80Å with 1Å resolution. The resulting distribution was normalized, a pseudo-count of 1×10^{-8} was added to each bin, and then the distribution was renormalized (this ensures that no bins have zero count and ensures that the maximum energy is bounded). The energy was then taken as $-k_B T \ln \rho(x)$, where $\rho(x)$ is the normalized distribution. This gives a broad distribution that takes into account uncertainty from both measurement error and probe flexibility. The resulting energy term varies from 0 to ~ 40 kJ mol^{-1}.

Restraints were scaled sequentially in the interval $\alpha \in [0.5, 1.0]$, so that short distances were added first (at higher α) and long distances later (α closer to 0.5).

Uncertain: this information comes from residue–residue contacts predicted from sequence evolution[15]. We take the top $N_{residues}$ predicted contacts from the EvFold server (http://evfold.org/evfold-web/evfold.do). The predicted contact could be any two atoms from the two residues involved in the contact being close in space. However, for computational efficiency we map this contacts onto the possible combinations of C_β and C_α pairs. This yields groups with four

possible combinations (two when involving one glycine and one when involving two glycines), one of which is active. The groups belong to a collection in which the active fraction is set to 0.80. Individually, each distance restraint has a flat bottom harmonic restraint with no restraint up to 6Å,the energy then increases quadratically up to 10Å and linearly afterwards. All restraints were scaled from full-strength to zero over the interval $\alpha \in [0, 1]$. Likewise, temperatures were scaled geometrically between 300 and 450K in the range of $\alpha \in [0, 1]$.

6 Force discontinuities

MELD uses sorting to determine the activity of each restraint in a group (or each group in a collection). This causes one or more forces to be instantaneously switched off at the same time others switch on. The construction of the energy function guarantees that the energy will be continuous. But the force will not: it has "cusps" when the active restraints change (see Figure 1 in the main text). These discontinuities cannot be integrated accurately; each time the set of active restraints changes, an integration error occurs.

In this section, we evaluate the effects of these errors. We use a simple test system, which we expect to be more pathological than our production systems. We assess the ability of MELD to produce correct results in three ways: (1) energy conservation, (2) expected temperature dependence of energy distributions, and (3) comparison with exact Monte Carlo simulations. We find that MELD produces the expected distributions.

6.1 Test system

Our test system consists of three carbon-mass particles confined to a $5 \times 5 \times 5$ Å^3 region of the non-periodic simulation box using a flat-bottomed harmonic potential with a force constant of 1000 kJ mol^{-1} nm^{-2}. In tests without MELD restraints, these particles do not interact at all—i.e. they are an ideal gas. In tests with MELD restraints, we add three restraints: between particles A and B, between B and C, and between A and C. The restraints have zero energy from zero to 1 Å, quadratic energy from 1 to 2 Å, and linear energy beyond 2 Å, with a force constant of 250 kJ mol^{-1} nm^{-2}. These parameters mimic typical values used for the protein systems in this study. We place all three restraints in a single group with one restraint active.

Due to the small volume and frequent collisions, the active restraint changes frequently, causing an integration error each time. In our protein systems, the active restraints change less frequently as the protein spends long periods of time trapped in a single basin defined by a set of restraints. This is especially true at low temperatures. Therefore, we expect this simple test system to be more sensitive to integration errors.

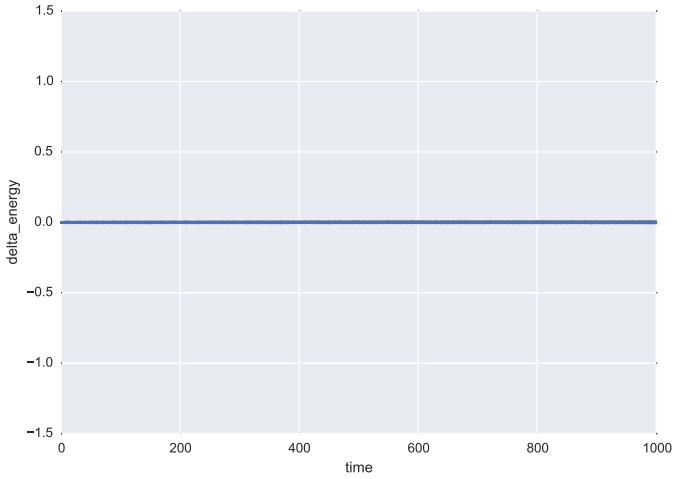

Figure S13: Non-thermostatted simulations without MELD restraints conserve energy.

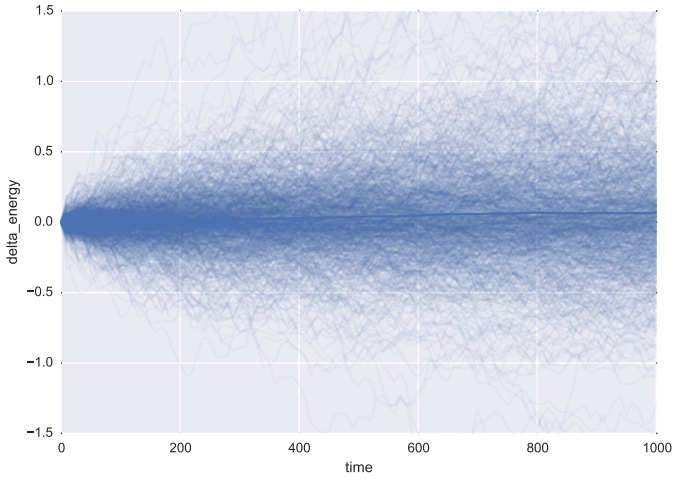

Figure S14: Non-thermostatted simulations with MELD restraints undergo random energy drift with a small systematic error. Each faint trace is a separate simulation. The darker line is the mean drift.

6.2 Energy conservation

We assessed energy conservation by running 1000 short (1 ns) simulations of our three-particle test system. These simulations used a Verlet integrator without temperature coupling, with a 2 fs timestep, and were started from random initial velocities drawn from a Maxwell-Boltzmann distribution at 300 K.

Simulations without MELD restraints conserve energy (Figure S13). All simulations bunch tightly together with an average drift of ($1.1 \times 10^{-5} \pm 6 \times 10^{-6}$) kJ mol^{-1} dof^{-1} ns^{-1}, comparable to other GPU simulation codes[16].

Simulations with MELD restraints (Figure S14) display far more variability in the energy, with individual simulations drifting as much as 2 kJ/mol during a 1 ns run. Importantly, however, the systematic drift is small. At each exchange, a small random error occurs with approximately zero mean error. The average energy drift with MELD restraints is $7 \times 10^{-3} \pm 2 \times 10^{-3}$ kJ mol^{-1} dof^{-1}ns^{-1}.

We expect that the error introduced from force discontinuities in the MELD potential will be less problematic than those introduced by truncating electrostatic or Lennard-Jones interactions. There are two reasons for this. First, the integration errors with truncated non-bonded interactions typically occur many times (between different pairs of atoms) per time step. In contrast, errors occur with MELD restraints only when the system crosses between energy basins defined by different sets of restraints, which is infrequent. Second, the integration error introduced by truncation of non-bonded interactions is systematic and typically leads to rapid heating of the system. Temperature coupling algorithms remove this excess heat, but this creates a non-equilibrium steady-state rather than the desired equilibrium distribution. In MELD the average error is close to zero, so we expect that the sampled distribution to be close to the desired distribution.

6.3 Ensemble preservation

Although the energy drift increases with MELD restraints, it is not clear what effect this will have on ensemble properties for thermostatted simulations. To test this, we applied a simple quantitative test developed by Shirts[17]. We simulated the system (either with or without MELD restraints) at two temperatures: 300 and 310 K for 100 ns using a Langevin thermostat with a 2 fs time step and $\gamma = 1.0\ ps^{-1}$. Using the `checkensemble` tools provided by the Shirts lab[17], we then assessed if the energy distributions $\rho(E_{\text{total}})$ were consistent with the temperature difference. For both the system with (Figure S15) and without (Figure S16) MELD restraints, the predicted slopes are within statistical error of the expected slope, deviating by 0.08 and 0.69 standard deviations for the simulations with and without MELD restraints, respectively. Thus, although the force discontinuities due to the MELD restraints introduce small, random integration errors, for simulations using a thermostat, the resulting energy distribution is statistically indistinguishable from a Boltzmann distribution.

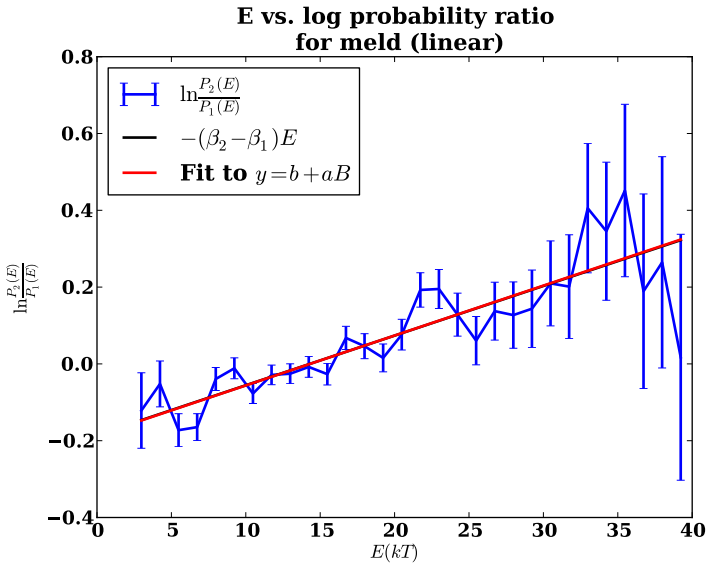

Figure S15: Systems with MELD restraints have log-ratios of energy distributions that have the expected slope to within statistical uncertainty. The fit slope is 0.09 standard deviations from the expected slope.

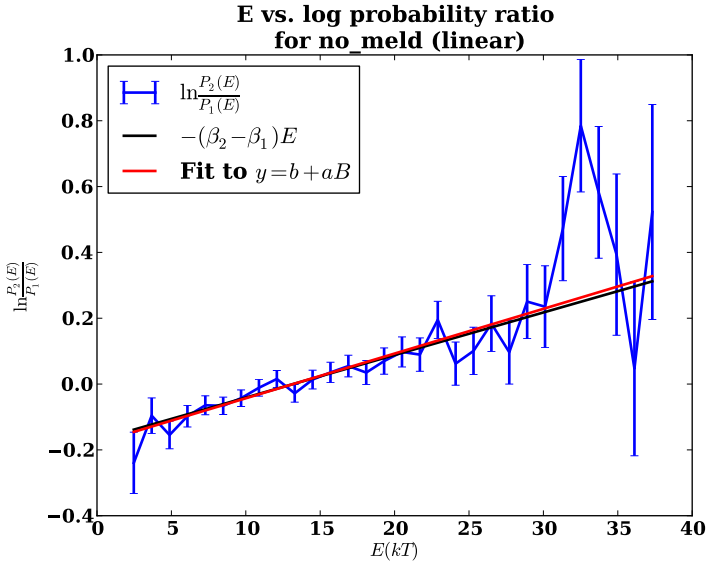

Figure S16: Systems without MELD restraints have log-ratios of energy distributions that have the expected slope to within statistical uncertainty. The fit slope is 0.69 standard deviations from the expected slope.

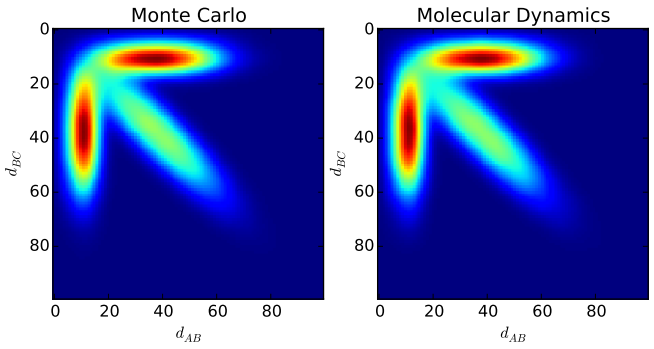

Figure S17: MELD produces results identical to exact Metropolis Monte Carlo simulations. d_{AB} and d_{BC} are the distances between particles A and B, or B and C, respectively.

6.4 Comparison with exact Monte Carlo results

To further verify that MELD produces the correct distributions, we compared the results with those from a simple Metropolis Monte Carlo (MMC) simulation. MMC does not use forces, and is thus immune to force discontinuities.

We performed 7.5×10^7 steps of either MD or MMC at a temperature of 300K. For MD, we used a Langevin integrator as described in the last section. For MMC, we perturbed the position of each particle by drawing from a gaussian distribution with zero mean and a standard deviation of 0.05 nm. Moves were accepted or rejected according to the standard Metropolis rule.

Figure S17 shows the joint distribution of A–B and B–C distances, using a kernel density estimate produced by `scipy.stats.guassian_kde` method with default parameters. Visually, the two distributions are indistinguishable. As a quantitative comparison, we compared the two distributions using the `scipy.stats.entropy` function and find the relative entropy to be < 0.001 bits.

This result strongly indicates that although there are occasional integration errors, MELD produces the expected conformational distributions.

6.5 Summary and alternatives

We expect that the discontinuities in the MELD restraints will not have a detrimental effect because: (1) the system crosses between restraint basins only rarely; (2) the average integration error is small and is not strongly biased towards heating or cooling the system; (3) tests on a model system show that with a thermostat, the distribution is statistically indistinguishable from a Boltzmann distribution; and (4) the results are indistinguishable from exact Monte Carlo results.

In some cases, for example certain types of minimization or integration schemes, it may be desirable or necessary to have continuous forces. This is easily achieved, but with an increase in computational complexity. One can replace the sorting operations in MELD with continuous analogues (e.g. using a sigmoidal functional form), although this entails an increase in computational complexity from $O(n \ln n)$ to $O(n^2)$, where n is the number of restraints.

7 Convergence of REMD

A replica exchange simulation is not converged until all replicas are converged[18]. One way to assess convergence is to examine the distributions sampled by each "walker" as it moves both within conformational space and within the replica exchange ladder[19]. Figure S18 shows the results for our Ubiquitin simulations, where it is clear that the walker distributions have not yet converged (the results for other systems are similar). This lack of convergence precludes a quantitative analysis of cluster populations (e.g. to get free energies). However, the most populous clusters nearly always contain an accurate model for all systems we have studied here. So, even though we cannot quantitatively analyze the cluster populations, the resulting structures are nevertheless useful predictions the native structure. We are actively exploring ways to improve the convergence, including optimized versions of replica exchange[19] and alternative sampling techniques like metadynamics[20, 21]. We expect that longer simulations and the use of improved sampling techniques will potentially lead to: (1) better models, as it is possible that MELD has not sampled the lowest-free-energy models; (2) more accurate identification of the best models, as the cluster populations are currently not converged and are thus only semi-quantitative; and (3) a better picture of the underlying structural preference of the force field and experimental data as the cluster populations become more converged.

8 Baseline XPLOR protocol

The following script was adapted from XPLOR-NIH tutorial material. The script was run 200 times, with the input data varied depending on the protein system under investigation.

```
xplor.requireVersion("2.24")


#
# slow cooling protocol in torsion angle space for protein G. Uses
# NOE, J-coupling restraints.
#
# this script performs annealing from an extended structure.
# It is faster than the original anneal.py
#
# CDS 2009/07/24
#
```

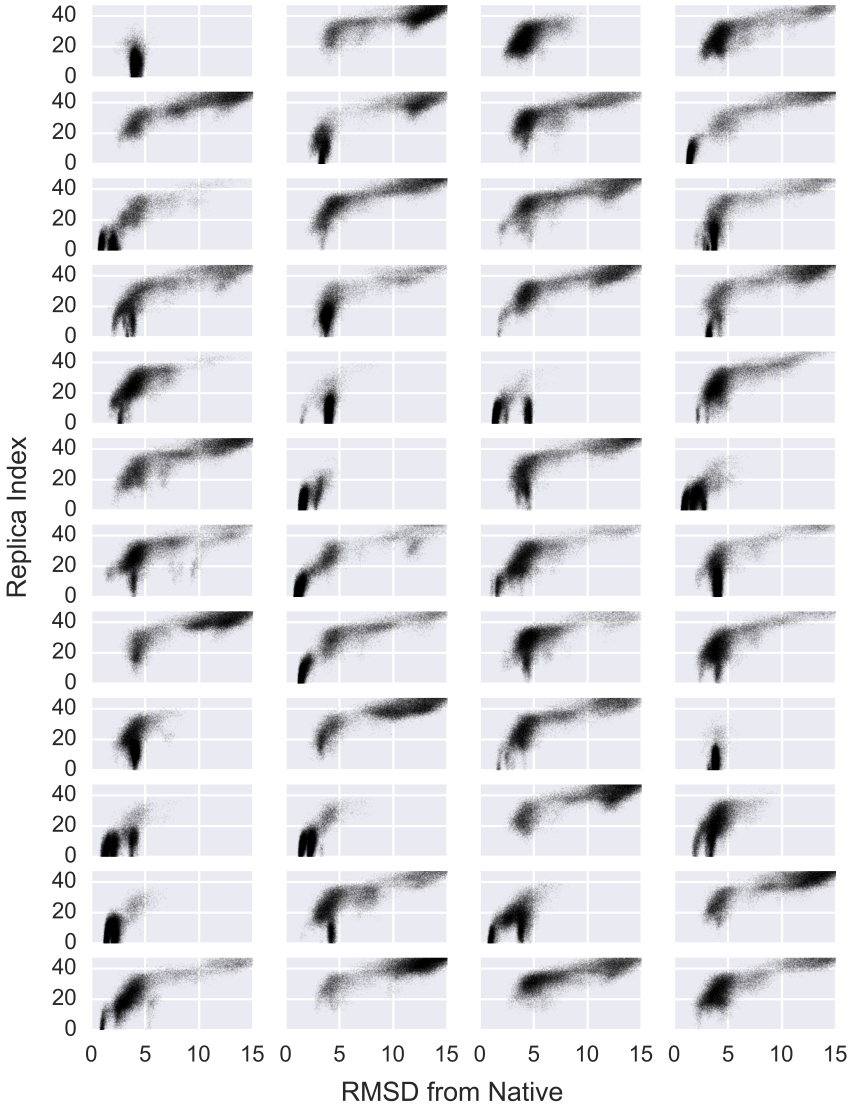

Figure S18: Distributions sampled by all 48 walkers in REMD simulations of Ubiquitin.

```
# this checks for typos on the command-line. User-customized arguments can
# also be specified.
#
xplor.parseArguments()


# filename for output structures. This string must contain the STRUCTURE
# literal so that each calculated structure has a unique name. The SCRIPT
# literal is replaced by this filename (or stdin if redirected using <),
# but it is optional.
#
outFilename = "SCRIPT_STRUCTURE.sa"
numberOfStructures=1    #usually you want to create at least 20

# protocol module has many high-level helper functions.
#
import protocol

import random
protocol.initRandomSeed(int(random.uniform(0.0, 1e7)))    #set random seed - by time

command = xplor.command

# generate PSF data from sequence and initialize the correct parameters.
#
from psfGen import seqToPSF
seqToPSF('sequence.seq')

# generate random extended initial structure with correct covalent geometry
#
protocol.genExtendedStructure()

#
# a PotList contains a list of potential terms. This is used to specify which
# terms are active during refinement.
#
from potList import PotList
potList = PotList()

# parameters to ramp up during the simulated annealing protocol
#
from simulationTools import MultRamp, StaticRamp, InitialParams

rampedParams=[]
highTempParams=[]

# set up NOE potential
noe=PotList('noe')
potList.append(noe)
from noePotTools import create_NOEPot
for (name,scale,file) in [('all',1,"dist.tbl"),
                          #add entries for additional tables
                          ]:
    pot = create_NOEPot(name,file)
    # pot.setPotType("soft") # if you think there may be bad NOEs
    pot.setScale(scale)
    noe.append(pot)
```

```
rampedParams.append( MultRamp(2,30, "noe.setScale(VALUE)") )

# Set up dihedral angles
from xplorPot import XplorPot
dihedralRestraintFilename="dihed.tbl"
protocol.initDihedrals(dihedralRestraintFilename,
                       #useDefaults=False  # by default, symmetric sidechain
                                           # restraints are included
                       )
potList.append( XplorPot('CDIH') )
highTempParams.append( StaticRamp("potList['CDIH'].setScale(10)") )
rampedParams.append( StaticRamp("potList['CDIH'].setScale(200)") )
# set custom values of threshold values for violation calculation
#
potList['CDIH'].setThreshold( 5 )



# gyration volume term
#
from gyrPotTools import create_GyrPot
gyr = create_GyrPot("Vgyr",
                    "resid 1:56") # selection should exclude disordered tails
potList.append(gyr)
rampedParams.append( MultRamp(.002,1,"gyr.setScale(VALUE)") )

# hbdb - hbond database-based term
#
protocol.initHBDB()
potList.append( XplorPot('HBDB') )

#New torsion angle database potential
#
from torsionDBPotTools import create_TorsionDBPot
torsionDB = create_TorsionDBPot('torsionDB')
potList.append( torsionDB )
rampedParams.append( MultRamp(.002,2,"torsionDB.setScale(VALUE)") )

#
# setup parameters for atom-atom repulsive term. (van der Waals-like term)
#
potList.append( XplorPot('VDW') )
rampedParams.append( StaticRamp("protocol.initNBond()") )
rampedParams.append( MultRamp(0.9,0.8,
                              "command('param nbonds repel VALUE end end')") )
rampedParams.append( MultRamp(.004,4,
                              "command('param nbonds rcon VALUE end end')") )
# nonbonded interaction only between CA atoms
highTempParams.append( StaticRamp("""protocol.initNBond(cutnb=100,
                                  rcon=0.004,
                                  tolerance=45,
                                  repel=1.2,
                                  onlyCA=1)""") )


potList.append( XplorPot("BOND") )
potList.append( XplorPot("ANGL") )
```

```
potList['ANGL'].setThreshold( 5 )
rampedParams.append( MultRamp(0.4,1,"potList['ANGL'].setScale(VALUE)") )
potList.append( XplorPot("IMPR") )
potList['IMPR'].setThreshold( 5 )
rampedParams.append( MultRamp(0.1,1,"potList['IMPR'].setScale(VALUE)") )



# Give atoms uniform weights, configure bath/molecule friction coeff.
#
protocol.massSetup()


# IVM setup
#   the IVM is used for performing dynamics and minimization in torsion-angle
#   space, and in Cartesian space.
#
from ivm import IVM
dyn = IVM()

# configure ivm topology for torsion-angle dynamics
#
protocol.torsionTopology(dyn)

# minc used for final cartesian minimization
#
minc = IVM()
protocol.initMinimize(minc)

protocol.cartesianTopology(minc)



# object which performs simulated annealing
#
from simulationTools import AnnealIVM
init_t  = 3500.      # Need high temp and slow annealing to converge
cool = AnnealIVM(initTemp =init_t,
                 finalTemp=25,
                 tempStep =12.5,
                 ivm=dyn,
                 rampedParams = rampedParams)


def calcOneStructure(loopInfo):
    """ this function calculates a single structure, performs analysis on the
    structure, and then writes out a pdb file, with remarks.
    """

    # generate a new structure with randomized torsion angles
    #
    from monteCarlo import randomizeTorsions
    randomizeTorsions(dyn)
    protocol.fixupCovalentGeom(maxIters=100,useVDW=1)

    # set torsion angles from restraints
    #
```

```
from torsionTools import setTorsionsFromTable
setTorsionsFromTable(dihedralRestraintFilename)
protocol.writePDB(loopInfo.filename()+".init")

# calc. initial tensor orientation
#

# initialize parameters for high temp dynamics.
InitialParams( rampedParams )
# high-temp dynamics setup - only need to specify parameters which
#   differfrom initial values in rampedParams
InitialParams( highTempParams )

# high temp dynamics
#
protocol.initDynamics(dyn,
                      potList=potList, # potential terms to use
                      bathTemp=init_t,
                      initVelocities=1,
                      finalTime=100,   # stops at 800ps or 8000 steps
                      numSteps=1000,   # whichever comes first
                      printInterval=100)

dyn.setETolerance( init_t/100 )  #used to det. stepsize. default: t/1000
dyn.run()

# initialize parameters for cooling loop
InitialParams( rampedParams )


# initialize integrator for simulated annealing
#
protocol.initDynamics(dyn,
                      potList=potList,
                      numSteps=100,        #at each temp: 100 steps or
                      finalTime=.2 ,        # .2ps, whichever is less
                      printInterval=100)

# perform simulated annealing
#
cool.run()


# final torsion angle minimization
#
protocol.initMinimize(dyn,
                      printInterval=50)
dyn.run()

# final all- atomic degrees of freedom minimization
#
protocol.initMinimize(minc,
                      potList=potList,
                      dEPred=10)
minc.run()

#do analysis and write structure when this function retunrs
```

```
    pass


from simulationTools import StructureLoop, FinalParams
StructureLoop(numStructures=numberOfStructures,
              doWriteStructures=True,
              pdbTemplate=outFilename,
              structLoopAction=calcOneStructure,
              genViolationStats=True,
              averageTopFraction=0.5, #report stats on best 50% of structs
              averageSortPots=[potList['BOND'],potList['ANGL'],potList['IMPR'],
                               noe,potList['CDIH']],
              averageContext=FinalParams(rampedParams),
              averageFilename="SCRIPT_ave.pdb",
              averagePotList=potList).run()
```

References

[1] Güntert P, Mumenthaler C, Wüthrich K (1997) Torsion angle dynamics for NMR structure calculation with the new program DYANA. *J Mol Biol* 273(1):283–298.

[2] Shen Y, Delaglio F, Cornilescu G, Bax A (2009) TALOS+: a hybrid method for predicting protein backbone torsion angles from NMR chemical shifts. *J Biol NMR* 44(4):213–223.

[3] Huber M et al. (2011) A proton-detected 4D solid-state NMR experiment for protein structure determination. *Chem Phys Chem* 12(5):915–918.

[4] Fukunishi H, Watanabe O, Takada S (2002) On the Hamiltonian replica exchange method for efficient sampling of biomolecular systems: Application to protein structure prediction. *J Chem Phys* 116(20):9058.

[5] Shao J, Tanner SW, Thompson N, Cheatham TE (2007) Clustering Molecular Dynamics Trajectories: 1. Characterizing the Performance of Different Clustering Algorithms. *J Chem Theo Comp* 3(6):2312–2334.

[6] Daura X, Gademann K, Jaun B (1999) Peptide Folding: When Simulation Meets Experiment. *Angew Chem Int Edit* 38(1/2):236–240.

[7] Nilsson L, Clore GM, Gronenborn AM, Brunger AT, Karplus M (1986) Structure refinement of oligonucleotides by molecular dynamics with nuclear overhauser effect interproton distance restraints: Application to 5 d(C-G-T-A-C-G)2. *J Mol Biol* 188(3):455–475.

[8] Eastman P et al. (2013) OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *J Chem Theo Comp* 9(1):461–469.

[9] MacKerell AD, Feig M, Brooks CL (2004) Extending the treatment of backbone energetics in protein force fields: Limitations of gas-phase quantum mechanics in reproducing protein conformational distributions in molecular dynamics simulations. *J Comp Chem* 25(11):1400–1415.

[10] Onufriev A, Bashford D, Case DA (2004) Exploring protein native states and large-scale conformational changes with a modified generalized born model. *Proteins* 55(2):383–394.

[11] Jones DT (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol*.

[12] Hirst SJ, Alexander N, McHaourab HS, Meiler J (2011) RosettaEPR: an integrated tool for protein structure determination from sparse EPR data. *J Struct Biol* 173(3):506–514.

[13] Alexander N, Bortolus M, Al-Mestarihi A, Mchaourab H, Meiler J (2008) De novo high-resolution protein structure determination from sparse spin-labeling EPR data. *Structure* 16(2):181–195.

[14] Islam SM, Stein RA, Mchaourab HS, Roux B (2013) Structural Refinement from Restrained-Ensemble Simulations Based on EPR/DEER Data: Application to T4 Lysozyme. *J Phys Chem B* 117(17):4740–4754.

[15] Marks DS et al. (2011) Protein 3D structure computed from evolutionary sequence variation. *PLoS One*.

[16] Case DA, al e (2012) *Amber12*. (University of California San Francisco).

[17] Shirts MR (2013) Simple Quantitative Tests to Validate Sampling from Thermodynamic Ensembles. *J Chem Theo Comp* 9(2):909–926.

[18] Roe DR, Bergonzo C, Cheatham III TE (2014) Evaluation of Enhanced Sampling Provided by Accelerated Molecular Dynamics with Hamiltonian Replica Exchange Methods. *J Phys Chem B* 118(13):3543–3552.

[19] Bergonzo C et al. (2014) Multidimensional Replica Exchange Molecular Dynamics Yields a Converged Ensemble of an RNA Tetranucleotide. *J Chem Theo Comp* 10(1):492–499.

[20] Laio A, Parrinello M (2002) Escaping free-energy minima. *P Natl Acad Sci USA* 99(20):12562–12566.

[21] Jiang P, Yaşar F, Hansmann UHE (2013) Sampling of Protein Folding Transitions: Multicanonical Versus Replica Exchange Molecular Dynamics. *J Chem Theo Comp*.